STATE-OF-THE-ART IN HIGH SPEED ARITHMETIC INTEGRATED CIRCUITS

Use of bipolar technology to construct arithmetic ICs has resulted in devices with increased switching speed and gate density and low power dissipation. Future technological advances should have an even greater impact through larger chip diameters and sharper pattern fabrication

Shlomo Waser Monolithic Memories, Incorporated, Sunnyvale, California

Current integrated circuit technology provides a full spectrum of arithmetic devices, varying in speed and capabilities. These devices can be divided into three groups according to their speed. In the first, or slowest, group are calculator chips that operate at greater than 1 ms; in the second, or medium speed group (1 μs and slower), n-channel metal-oxide semiconductor microprocessors; and in the third, or fastest group (1 μs or better), bipolar data slices and bipolar discrete units (adders, multipliers, etc). The third group of high speed arithmetic integrated circuits, under discussion here, are significant building blocks in constructing "number crunching" systems for use in weather modeling, nuclear physics computations, and realtime digital signal processing tasks such as speech and image processing, computerized tomography, and air traffic monitors.

Technology Background

To understand the difficulties encountered in fabricating high speed arithmetic integrated circuits (ICs), it is necessary to review semiconductor technology in general, with particular attention to bipolar technology. In this respect, consider the three factors that limit how much parallel arithmetic can be put into one chip: maximum allowed power dissipation, pin count, and cost.

Maximum Allowed Power Dissipation

In high speed technologies (mostly bipolar), maximum number of gates is a direct function of the chip's maximum allowed power dissipation. This, in turn, depends on maximum allowed junction temperature of the silicon die, ambient temperature, and ability of the IC package to dissipate heat (thermal resistance). This relationship is defined as

$$\text{Max power dissipation} = \frac{T_{\text{junction}} - T_{\text{ambient}}}{\theta_{\text{ja}}}$$

For military specifications, $T_{\text{ambient}} = 125$ °C, a typical package at still air has a thermal resistance (θ_{ja}) of 40 °C/W, and the maximum allowed junction temperature is 175 °C [for transistor-transistor logic (TTL)]. These values give 1.25-W maximum power dissipation, which is typical of most large-scale integrated (LSI) devices on the market today. However, effective thermal resistance between the IC package and the ambient environment can be reduced to 15 °C/W by attaching the package to a heat sink and cooling it by forced air.[1] If IC operation is limited to the commercial temperature range (maximum 70 °C), maximum power dissipation is about 7 W. While this latter dissipation value seems feasible, available ICs (with this characteristic) are undesirable, because they will usually be interfaced and surrounded by ICs that dissipate less than 1 W. Thus, a local hot spot is generated that is difficult to cool efficiently and reliably.

The maximum number of gates that can be integrated into one chip is equal to the maximum allowed power dissipation divided by the power dissipation of each gate. For example, one of the most popular and successful technologies today is low power Schottky TTL (LS/TTL) which dissipates 2 mW/gate; this LSI chip type contains about 600 gates.

In a given technology, power dissipation of each gate is roughly proportional to its physical size, which in turn is determined by number of active elements and resolution of the lithography used to define the geometries of the transistors. For random-access memories (RAMs) the progress from 1k to 16k bits was mostly due to reducing the 3-transistor cell to a 1-transistor cell and cutting the line width of the fabrication pattern from 10 to 5 μm. Introduction of the 64k RAM will probably require an improvement over present photolithography techniques, with electron beam lithography the most likely process.

Pin Count

Number of available pins in the IC package is a severe limitation for high speed arithmetic on wide words. Multiplying 16-bit operands requires a package with at least 66 pins. The state-of-the-art is 64 pins, and no major breakthrough is in sight[2]. Even custom ICs do not exceed this limit by much, although the Amdhal computer uses a custom 84-pin flat-pack package. Pin count limitations have led to the data "slice" concept; ie, partitioning the desired system into identical parts that have common control and interconnection mechanisms (carry, etc). The pin limitation can also be circumvented by time-multiplexing the information via common pins; however this reduces the overall speed.

Cost

If power dissipation is not a limiting factor, as with metal-oxide semiconductor (MOS) and integrated-injection logic (IIL) technologies, cost will limit the die size. Noyce[3] points out that if yield is a function of random defects, cost increases exponentially with die size. For example, if a given chip size yields 10% good die, a chip twice as large will yield 1%. The cost for twice the function will be 20 times as great. Other major cost elements are testing and assembly, which are approximately fixed per chip. Consequently, as the number (N) of functions per chip increases, the assembly cost decreases proportionally to 1/N. Minimum cost per function will be at the crossover point between silicon chip cost and assembly and test cost. Progress in ICs can be viewed as a fixed cost (about $10) for increased complexity (three orders of magnitude in the last 15 years).[2]

Characterizing Technologies

To simplify comparisons among the various technologies, the "natural" gate implementation is analyzed. Natural gate is a realization of a Boolean operator that requires a minimum number of transistors while giving maximum speed. For TTL, this gate is a NAND; for ECL, it is a NOR. Unfortunately, no known technology has the exclusive-OR as its natural gate. Unless otherwise stated, the word gate will be used instead of natural gate.

Two main characteristics of a gate are power and speed. A common figure of merit is the speed-power product, where speed is in nanoseconds (ns), power is in milliwatts (mW), and product is in picojoules (pJ). A third characteristic is the gate's fan-in. In most bipolar technologies, fan-in is four. In the TTL family, NAND gates with one, two, three, or four inputs have the same speed and the same power, whereas larger fan-in NAND gates have larger speed-power products. Limited fan-in is a severe limitation in arithmetic operations on wide words requiring carry-lookahead (CLA).

Bipolar Technologies

All fast IC technologies are bipolar; the two most commonly used today for high speed arithmetic hardware are ECL and Schottky-TTL. Integrated-injection logic (IIL) is a relatively new technology that will become important for very large-scale integration (above 1000 gates/chip).

Emitter-Coupled Logic

ECL is the fastest, commercially available technology; gate delay is 1 to 2 ns, and speed-power product is 50 pJ. Although introduced about 10 years ago, the technology is still limited to applications that require very high speed, such as top of the line mainframe computers (IBM 370/168, Amdhal 470, DECsystem 10) and some signal processing equipment because of design difficulties. Interconnections become, in effect, transmission lines that require proper termination and matching. Also, reliable ECL designs need to use multilayer printed circuit (PC) boards, which are expensive. Since ECL devices normally are powered from a —5.2-V supply, they are incompatible with the popular TTL family. Another severe shortcoming of the ECL devices is their excessive power dissipation of 25 to 60 mW/gate, which requires forced air cooling. Nevertheless, if maximum speed is needed, ECL is the best choice. The internal circuit of ECL is a current-switching mechanism, which implies a constant current drain of the power supply. By contrast, the TTL logic family, using voltage threshold, causes large current spikes on the power supply during switching from one state to another.

ECL technology is implemented by various families. The fastest (1 ns) is MECL III from Motorola,[4] Fairchild, and Signetics. The 10,000 family has the largest selection of ECL LSI devices, and it is the only ECL family with arithmetic units. A typical gate operates at 2 ns and 25 mW.

Transistor-Transistor Logic

TTL technology was introduced in 1964 by Texas Instruments (TI).[5] It has been the most popular logic family for more than a decade. The original family had a 10-ns gate at 10-mW dissipation, and devices of this family are still the least expensive ICs. However, two subfamilies are making inroads. One is the Schottky-TTL (S/TTL) with a 3-ns gate at 20 mW, matching the speed-power product

TABLE 1

Comparison of Common Bipolar Technologies Used in Implementing High Speed Arithmetic Devices

		Gate Characteristics				2-Input Exclusive-OR			
Technology/ Year Introduced	Function	Delay	Power	Speed-Power Product	Density (gates/mm²)	Delay	Power	Speed-Power Product	Comments
ECL-III (1968)	NOR	1.1 ns	60 mW	66 pJ	30	1.3 ns	70 mW	91 pJ	Limited number of functions
ECL-1000 (1971)	NOR	2 ns	25 mW	50 pJ	30	2.5 ns	50 mW	125 pJ	Large selection of functions
S/TTL (1970)	NAND	3 ns	20 mW	60 pJ	30	7 ns	60 mW	420 pJ	Large selection of functions
LS/TTL (1972)	NAND	10 ns	2 mW	20 pJ	30	10 ns	8 mW	80 pJ	Large selection of functions
IIL (1975)	NAND	10 ns	0.1 mW	1 pJ	300				Not a mature technology
NMOS (1973)		100 ns	0.1 mW	10 pJ	130				For reference only
EEIC (1977)		0.25 ns	2 mW	0.5 pJ	100				Still in research and development

of ECL. The second subfamily, low power Schottky (LS/TTL), retains the speed of original TTL but decreases the power dissipation to 2 mW. Popularity of TTL technology has led to the largest selection of different ICs. Most small-scale integrated (SSI) and medium-scale integrated (MSI) devices are triplicated in three subfamilies (TTL, S/TTL, LS/TTL); however, most LSI devices are implemented only by LS/TTL. TTL ICs operate from a 5-V power supply. No critical problems have envolved in PC board layout as in ECL, and power dissipation typically does not require any special cooling. As noted previously, precautions need to be taken in decoupling the power supply lines, due to current spikes that are present while switching from one state to another.

Integrated-Injection Logic

IIL is a relatively new technology[6] (1975) that has not matured like ECL and TTL; thus, conflicting reports exist about its potential characteristics. Nevertheless, a gate with LS/TTL speed of 10 ns and power dissipation of 0.1 mW is reported.[7] It is likely that by 1980, most high density monolithic arithmetic processors will be implemented in IIL, replacing LS/TTL completely for such applications. In fact, the photochemical process used in fabricating LS/TTL can be modified to handle IIL, making it even more attractive than a completely new technology.

Most digital IIL devices are powered from a 5-V power supply to retain TTL compatibility. However, IIL technology needs only 1 V to operate, since it requires only current sourcing, and no voltage thresholds are used. Thus, for applications where TTL compatibility is not required, further power reduction is possible at no sacrifice in speed.

Table 1 summarizes the characteristics of the described technologies. A characterization of the implementation of an exclusive-OR gate is included for each technology because this Boolean operator is the major element in implementing digital arithmetic.

Arithmetic Elements

Arithmetic Logic Unit

Arithmetic logic units (ALUs)[4,8] are capable of add, subtract, shift, and logic operations (AND, OR, ex-OR). The most popular ALU device is the 74S181 implemented in S/TTL technology (or 74181), which is used in minicomputers such as the DEC PDP-11 and the Data General NOVA. This device performs addition using a carry-lookahead algorithm across four bits at a time. When operating on wider words, a companion device (74S182) provides a full carry-lookahead across any number of bits. Each carry-lookhead unit (74S182) receives the generate and propagate terms from a group of four 74S181s. In general, the number of levels of carry-lookahead is $\log_4 n$; eg, adding 64 bits with full carry-lookahead takes 15 ns in ECL and 28 ns in S/TTL.

The 74S181 (and the 10181) are combinatorial devices, and accumulation of results requires an additional register (accumulator). The 74S281 is an ALU with an accumulator on one chip, which still uses the 74S182 for carry-lookahead. Adding and storing 64-bit operands take 42 ns.

Texas Instruments has introduced two additional -81 devices. The 74S381 is similar to the 74S181—some functions of the 74S181 were removed to enable it to be packaged in a 20-pin, 0.3″ (7.6-mm) package instead of the 24-pin, 0.6″ (15.2-mm) package used to house the 74S181. The second device, the 74S481, will be discussed in the section dealing with processor elements.

Table 2 lists the speed and power of commonly used ALUs. As can be expected, power is approximately proportional to the gate count of the device. The first three devices listed in the table are basic ALU elements. 74S181 is implemented in S/TTL; the 10181 in ECL. 74S182 and 10179 serve as support chips for the ALU, and provide carry-lookahead when several ALUs are cascaded. The 74S281 contains an on-chip accumulator, but is slower than the combinatorial ALUs. Table 3 extends the chip comparison into the system level. When several ALUs are cascaded, the carry-lookahead units provide faster addition and subtraction, at the expense of increased power dissipation. S/TTL addition of 64 bits takes only 28 ns, but results in 11-W power dissipation. Addition of only 4 bits does not require carry-lookahead, addition of 16 bits requires only 1 carry-lookahead, but addition of 64 bits uses 5 carry-lookahead units.

Multipliers

In describing the ALU adder ICs it is clear that the most common addition algorithm is the carry-lookahead, and the most common configuration is four bits per IC. In contrast, multipliers use a variety of algorithms and configurations.

Parallel multiplication algorithms can be divided into two types: those that generate partial products and those that add the partial products. In generating partial products. The n bits of the ith partial product method is to use AND gates. If the multiplier (Y) and the multiplicand (X) each have n bits, there are n partial products. The n bits of the ith partial product are generated by ANDing Y_i with each of the n bits of the multiplicand X. However, in 2's complement representation, a correction is required since the most significant bit (MSB) has, effectively, a negative weight.

Booth's algorithm[9] is a method of recoding the multiplier so that the sign bit (MSB) is treated in the same way as the rest of the bits. A modified Booth's algorithm, suggested by MacSorley,[10] serves as a means of halving the number of partial products while keeping the elegance (sign bit treated as any bit) of the original algorithm for 2's complement numbers. The reduced number of partial products increases multiplication speed and decreases gate count. The motivation behind Booth's algorithm is to skip over a string of 1s and 0s, rather than form a partial product for each bit. Skipping a string of 0s is clear. Skipping over a string of 1s involves computing a string of 1s by subtracting the weight of the rightmost 1 from the modulus of the string. For example, the binary string 1111 is $2^4 - 2^0 = 15$, and the binary string 11100 is $2^5 - 2^2 = 28$.

In the actual hardware implementation, Booth's algorithm requires that the operand (multiplier) be divided into N/2 groups or substrings, each of which has

TABLE 2

Comparison of Arithmetic Logic Units

Part No.	Function	Gate Count	Speed	Power
74S181	4-bit ALU	75	11 ns	600 mW
10181	4-bit ALU	75	7 ns	600 mW
74S281	4-bit ALU/ accumulator	100	22 ns	700 mW
74S182	4 groups carry-lookahead	20	7 ns	350 mW
10179	4 groups carry-lookahead	20	4 ns	300 mW

TABLE 3

Comparison of Speed/Power at System Level

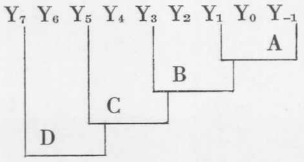
Part No.	Technology		Speed/Power	
		4 Bits	16 Bits	64 Bits
74S181/ 74S182	S/TTL	11 ns/600 mW	18 ns/2.7 W	28 ns/11 W
10181/ 10179	ECL	7 ns/600 mW	11 ns/2.7 W	17 ns/11 W

three bits. Assume that the multiplier has the binary pattern 0111. A 0 digit is added to the right, and the resultant number (01110) is divided into two 3-bit groups. All possible permutations of these substrings are computed from the chart to determine the partial products.

2^1	2^0	2^{-1}	
Y_{i+1}	Y_i	Y_{i-1}	
0	0	0	Add zero (no strings)
0	0	1	Add multiplicand (end of string)
0	1	0	Add multiplicand (a string)
0	1	1	Add twice the multiplicand (end of string)
1	0	0	Subtract twice the multiplicand (beginning of string)
1	0	1	Subtract the multiplicand (−2X + X)
1	1	0	Subtract the multiplicand (beginning of string)
1	1	1	Subtract zero (center of string)

The first group is 110, which requires subtracting the multiplicand; the second group (011) requires adding twice the multiplicand. Since the second group is shifted twice, its relative weight is four times that of the first group. Thus, "adding twice" for the second group means "adding eight times." Combining the two groups, the result is "add seven times the multiplicand." This method requires only six easy operations: ±0, ±X, ±2X.

Every two contiguous groups have one bit in common as follows.



This padded 8-bit multiplier is divided into four groups, each made up of three bits. Each of the groups is operated upon by encoding the previous definitions. However, each group has a different weight. Group A has a weight of 1, group B a weight of 4, and so on. Note that bit −1 is always 0.

Thus the modified Booth's algorithm is a multiplier encoding scheme that involves a constant shift of two bits at a time while examining three multiplier bits, resulting in N/2 partial products rather than the N partials involved without encoding. This algorithm can be extended by shifting three bits at a time while examining four bits at each subgroup. However, in encoding some permutations of a 4-bit string, such as 0110, the partial product is three times the multiplicand. Since generating a multiplication of three is not as trivial as the shifting used in generating a multiplication of two, none of the semiconductor multipliers use more than three bits for encoding.

The second type of multiplication algorithm deals with adding the partial products. All parallel algorithms use the carry-save adder; this adder is identical to a

TABLE 4

Comparison of Available Multiplier ICs

Vendor/ Device	Configuration	Pins	Speed	Power	Data Code	Amount of Parallelism	Algorithm
TRW/ MPY-8	8 x 8	40	130 ns	1.2 W	2's comp	Full	AND gates and carry-save adders
TRW/ MPY-12	12 x 12	64	150 ns	3.5 W	2's comp	Full	AND gates and carry-save adders
TRW/ MPY-16	16 x 16	64	180 ns	5 W	2's comp	Full	AND gates and carry-save adders
MMI/ 67558	8 x 8	40	100 ns	1 W	2's comp/ unsigned	Full	Modified Booth's; modified Wallace Tree
MMI/ 67516	16 x 16	24	800 ns	1 W	2's comp	Multiplicand/ 2 multiplier bits	Modified Booth's
MMI/ 67508	8 x 8	20	400 ns	0.75 W	2's comp	Multiplicand/ 2 multiplier bits	Modified Booth's
AMD/ 25S05	2 x 4	24	25 ns	0.6 W	2's comp	Full	Booth's
AMD/ 25LS14	8 x 1	16	50 ns	0.5 W	2's comp	Multiplicand/ 1 multiplier bit	Booth's
AMD/ 25LS2516	8 x 8	40	400 ns	1 W	2's comp	Multiplicand/ 2 multiplier bits	Modified Booth's
TI/ 74S274 (ROM)*	4 x 4	20	50 ns	0.5 W	Unsigned	Full	ROM lookup table*
Motorola/ 10183	2 x 4	24	20 ns	0.8 W	2's comp	Full	Booth's

*When used with companion device 74S275, partial products are added in a Wallace Tree configuration

TABLE 5

Performance Comparison Between 8 x 8 and 16 x 16 Multiplication

			8 x 8 Multiplication			16 x 16 Multiplication		
Vendor/ Number	Configuration	Pins	No. of Packages	Speed	Power	No. of Packages	Speed	Power
MMI/ 67558	8 x 8	40	1	100 ns	1 W	14*	140 ns	9 W
TRW/ MPY-8	8 x 8	40	1	130 ns	1.8 W	14*	170 ns	10 W
Motorola/ 10183	2 x 4	24	8	50 ns	6.4 W	32	100 ns	25.6 W
AMD/ 25S05	2 x 4	24	8	75 ns	5 W	32	150 ns	20 W
TI/ 74S274	4 x 4	20	12**	75 ns	5.4 W	45	120 ns	21 W
TRW/ MPY-16	16 x 16	64	—	—	—	1	180 ns	5 W
MMI/ 67516	16 x 16	24	—	—	—	1	800 ns	1 W
AMD/ 25LS2516	8 x 8	40	1	400 ns	1 W	2	800 ns	2 W

*4 packages are 8 x 8 multipliers, 10 are adders (74S181/10181)
**4 packages are 4 x 4 multipliers, 8 more are Wallace Tree bit-slices (74S275)

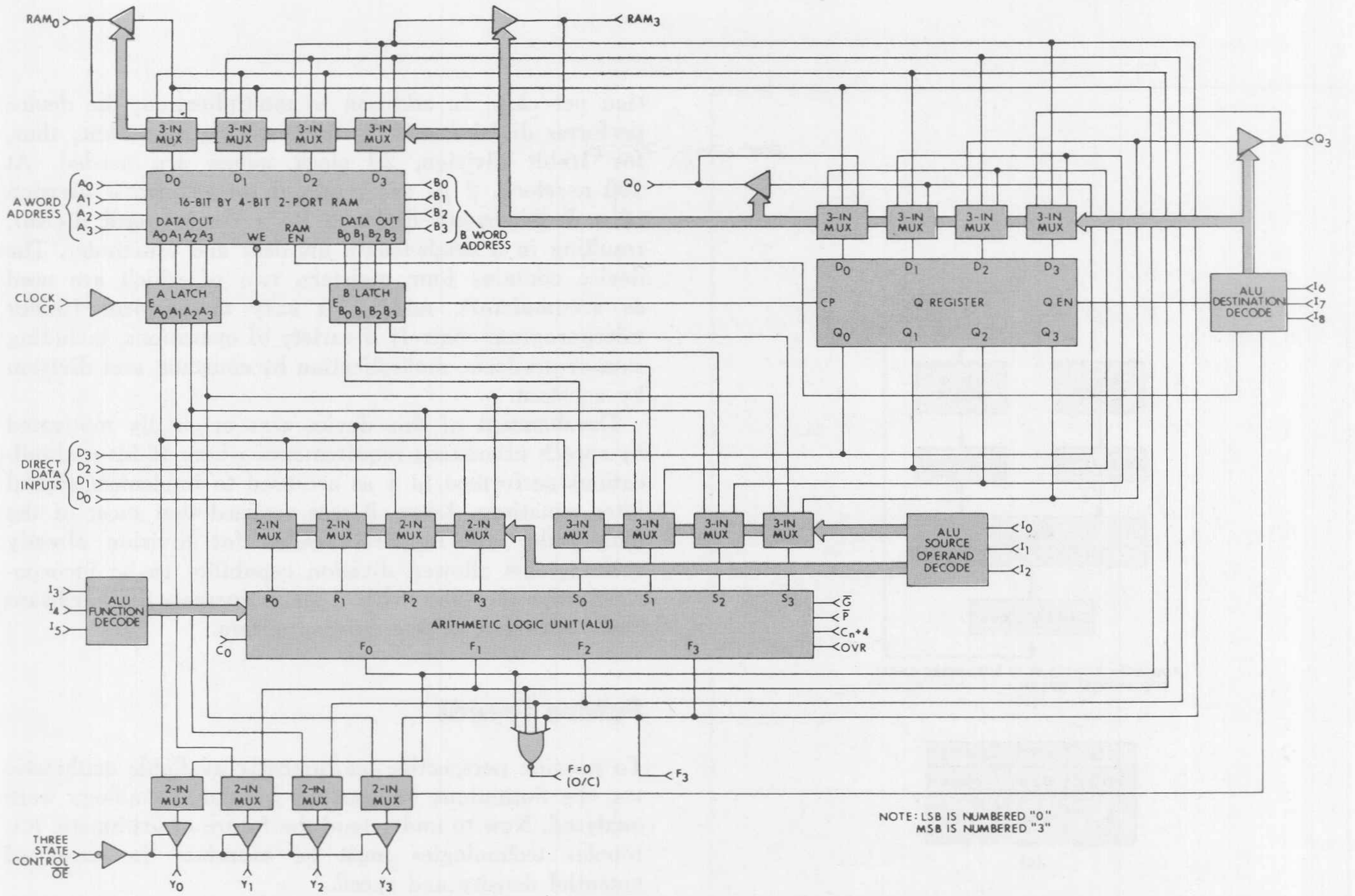
binary full adder. The only difference is in the interconnections of carries; instead of waiting for the carry to ripple, the carry is added at a later stage. Postponement of the addition of carries can be extended to all adder stages except the last. Carries from the last stage essentially form an n-bit operand to be added to the n-bit sum, and this operation can be done by carry-lookahead adders. Another side benefit of postponing addition of the carries to a later stage is the availability of a third input in each adder in the first stage; thus, the first three partial products can be added by the first stage, reducing the number of adder stages by one. This scheme is a minor modification of the Wallace Tree.[11]

Table 4 summarizes available IC multipliers and the algorithms that they use. For example, the TRW multipliers[12] use AND gates to generate partial products which are added by carry-save-adders, however (unlike the Wallace Tree), they let the carries ripple through the last adder stage.

The MMI 8 x 8 multiplier[13] (67558) generates partial products by using the modified Booth's algorithm. These partial products are then added in a Wallace Tree configuration. Texas Instruments extends the AND gate concept and provides a ROM (74S274) to generate a 4 x 4 segment of the partial products. They also provide a 7-bit Wallace Tree slice (74S275) for use in adding the partial products. This bit-slice can be used with other multipliers to provide an expanded multiplication. AMD 25S05[14] and the Motorola 10183 provide an onchip solution to the expansion problem by implementing X times Y plus K instead of just X times Y.

Some multipliers in Table 4 are semiparallel. The AMD 25IS14 generates and accumulates one partial product at each clock pulse; this product is made up of one multiplier bit and the full width of the multiplicand. Thus, for 8 x 8 multiplication, eight clock pulses are required. The MMI 67516 is similar except that it shifts two bits at a time and the multiplicand is 16-bits wide. Thus, 16 x 16 multiplication is performed in eight cycles.

Table 5 compares performance of the various chips in performing 8 x 8 and 16 x 16 multiplication, and can be used to make the engineering tradeoff in system design of multiplication. If maximum speed is needed, say for 8 x 8 multiplication, then using eight packages



NOTE: LSB IS NUMBERED "0"
MSB IS NUMBERED "3"

Fig 1 Detailed AM2901 microprocessor block diagram. Main element of 2901 is ALU, which performs eight operations according to three instruction lines. Source and destination of operands and results are determined by remaining six instruction lines. Register file, in upper left corner, is 16 x 4 dual-port RAM, which is used as 16 registers or accumulators for ALU. External input and output data buses are also used as source and destination, respectively

of the ECL 10183 is the best choice; however, if the associated power dissipation of 6.4 W is excessive, the single chip MMI 67558 is best. While it multiplies somewhat slower than the ECL device, its power dissipation is only 1 W.

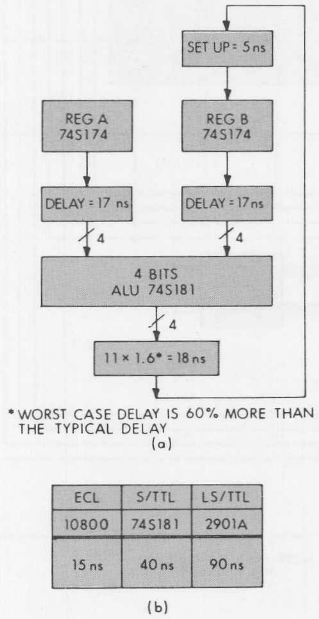
Bit-Slice Processor Elements

The first arithmetic IC was a binary adder, which was integrated later to the combinatorial ALU (74181). The next evolution of integration (after the ALU) received many different names, such as bit-slice microprocessor, RALU (ALU with registers), and data slice. The former is the most popular description. Architecture of the bit-slice is made typically from the classic ALU, but with multiple accumulators (registers) and control over ALU sources and destination. Fig 1 shows the architecture of the AMD-2901,[15] which is probably the most commonly used bit-slice processor element. Other such elements are Motorola MOT 10800, MMI 6701, Intel 3000, TI 74S481, and TI SBP 400. All have four bits per IC, except the Intel 3000 which is only two bits wide.

To determine the speed of these devices, the register to register operation ($R_A + R_B \rightarrow R_B$)* is examined. The 2901A, a faster version of the 2901, performs such an operation in 90 ns. To compare this speed with that of the 74181 type ALU, it is necessary to add registers to the 74181, as shown in Fig 2.

Expanding the bit slices to handle more than four bits is similar to expanding the 74181 ALUs. In fact, most bit-slice vendors recommend using the same carry-lookahead unit (74S182). To get a more realistic picture of the actual throughput of these devices, it is necessary to assume some overall system of architecture. Fig 3 shows the architecture for a 16-bit system. It is assumed to be microprogrammable with a pipelined microinstruction register; ie, maximum speed is achieved when no branch decisions are made. Fig 3 also contains a comparison of 16-bit throughput for various building blocks. In computing the addition time of the ALUs

*$R_A + R_B \rightarrow R_B$ means adding the contents of register A to the contents of register B and storing the sum into register B. All of this is accomplished in one cycle.



ECL	S/TTL	LS/TTL
10800	74S181	2901A
15 ns	40 ns	90 ns

Fig 2 MSI emulation of bit-slice processor element. (a) To compare simple ALU with bit-slice processor element, two registers are added to ALU. (b) Time to perform four bits ($R_A + R_B \rightarrow R_A$) is tabulated for various building blocks

(74S181 and 10181), the configuration assumed is similar to that of Fig 2, with the addition of a multiplexer (to select one of several sources) and a microinstruction register. From Fig 3(b) it can be seen that the speed of the arithmetic processor elements is about half that of the older ALUs. This ratio applies to both TTL and ECL technologies. Slower speed of these elements is probably the main reason that many recent minicomputers still use the older ALUs.

Monolithic Arithmetic Processors

The MMI 67516 (mentioned in the discussion on multipliers) continues the trend toward greater integration per chip. In addition to multiplication, the device performs division using a nonrestoring algorithm; thus, for 16-bit division, 20 clock pulses are needed. At 100 ns/clock, 2 μs are required to perform a division of a double-length dividend by a single-length divisor, resulting in a single-length quotient and remainder. The device contains four registers, two of which are used as accumulators, making it easy to perform (under microprogram control) a variety of operations, including sum of products, multiplication by constant, and division by constant.

Development of this device was originally motivated by speech processing requirements, where 16-bit multiplications performed in 1 μs are used to implement digital filter equations. Later, it was realized that most of the data paths and registers needed for division already existed; this allowed division capability to be incorporated into the chip with a small increase in hardware and expansion of the microprogram.

Future Trends

To provide perspective on currently available arithmetic ICs, the limitations imposed by bipolar technology were analyzed. Now to understand the future of arithmetic ICs, bipolar technologies must be examined in terms of potential density and speed.

Two major density improvements are in sight. The first is an increase of the wafer diameter from 3 to 4 in (7.6 to 10.1 cm). This increase, which has been implemented by some companies, doubles the chip size while maintaining the same cost (up to a limit, the cost of processing a wafer is almost independent of its diameter). The second, more revolutionary, improvement is a new method of drawing the patterns needed for the fabrication of integrated circuits. Resolution currently obtained with optical lithography is lines approximately 2 to 5 μm wide. Further resolution in optical techniques is limited by diffraction effects that occur between the mask and the wafer. The new method—electron-beam lithography—provides up to 20 times the resolution of optical lithography.[16]

With these improvements, Texas Instruments expects to achieve a chip size of 140,000 sq mils.[17] With IIL technology, this chip size will contain about 10,000 gates. At this high level of integration, many arithmetic ICs will probably give way to a single-chip high speed microprocessor that performs addition, multiplication, and division all at the same speed. Thus, one direction in the future will be further integration of multichip systems into a single chip while maintaining the same system speed.

A second possible future trend will be to retain the same level of integration but to employ higher speed technology. A new bipolar process—elevated electrode integrated circuit (EEIC)[18]—is reported to have 250-ps delay/gate at 2-mW power dissipation. With such a technology (still under research and development), it

(b)

ECL		S/TTL	LS/TTL	IIL
10181	10800	74S181	2901A	SBP400
32 ns	62 ns	68 ns	143 ns	300 ns

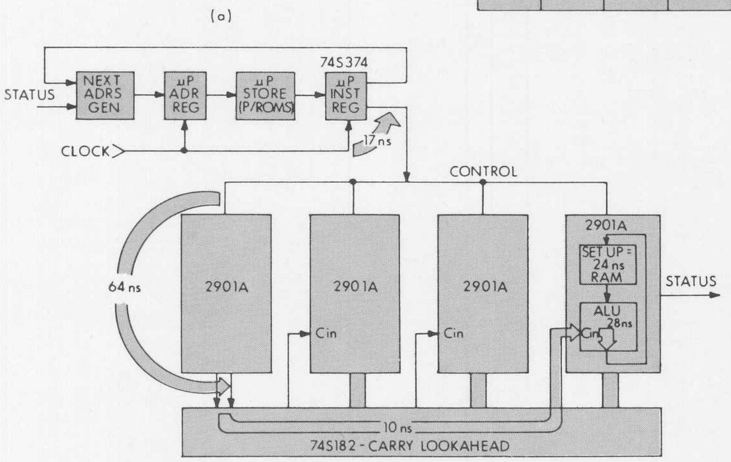


Fig 3 Architecture and performance of microprogrammable system. (a) Microprogrammable system (16 bits) made up of 2901s. For simplicity, data bus is not shown. (b) Comparison of minimum time to perform $R_A + R_B \rightarrow R_B$ for 16-bit systems made up of different building blocks

will be possible to construct an ALU that is functionally similar to the 74S181, but with an order of magnitude speed improvement, ie, 1 ns. A 16-bit computer made with this ALU and carry-lookahead units could perform register to register addition in 7 ns, if a sufficient variety of devices in the new family exist to build such a computer.

The future directions outlined are merely interpolations of the progress to date. As with the microprocessor revolution, it is possible that a completely new direction in computer arithmetic will emerge from the continuing advancements in IC technology.

References

1. J. Lyman, "Growing Pin Count is Forcing LSI Package Changes," *Electronics*, Mar 17, 1977, pp 81-91
2. D. A. Hodges, "Trends in Computer Hardware Technology," *Computer Design*, Feb 1976, pp 77-85
3. R. N. Noyce, "From Relays to MPUs," *Computer*, Dec 1976, pp 26-29
4. "MECL Data Book," Motorola, Phoenix, Ariz, 1974
5. R. L. Morris *et al*, *Designing with TTL Integrated Circuits*, McGraw-Hill Book Co, New York, NY, 1971
6. P. J. Verhofstadt, "Technology for Microprocessor Hardware," *Digest of Papers*, IEEE Computer Society International Conf, Feb 1976, pp 19-22
7. D. Wilnai, "LSI Elements Using I³L Technology," *Digest of Papers*, IEEE Computer Society International Conf, Feb 1977, pp 103-106
8. *The TTL Data Book for Design Engineers*, Texas Instruments, Inc, Dallas, Tex, 1976
9. A. D. Booth, "A Signed Binary Multiplication Technique," *Quarterly Journal of Mechanisms and Applied Mathematics*, Vol 4, Part 2, 1951
10. O. L. MacSorley, "High Speed Arithmetic in Binary Computers," *Proceedings of the IRE*, Jan 1961, pp 67-91
11. C. S. Wallace, "A Suggestion for a Fast Multiplier," *IEEE Transactions on Electronic Computers*, Feb 1964, pp 14-17
12. MPY-Series Multipliers," TRW, Redondo Beach, Calif, 1976
13. "67558 Data Sheet," Monolithic Memories, Inc, Sunnyvale, Calif, Aug 1977
14. "Digital Signal Processing Handbook," Advanced Micro Devices, Inc, Sunnyvale, Calif, 1976
15. *The AM2900 Family Data Book*, Advanced Micro Devices, Inc, Sunnyvale, Calif, 1976
16. T. H. P. Chang, *et al*, "Electron Beam Lithography Draws a Finer Line, "*Electronics*, May 12, 1977, pp 89-98
17. "Very Large Scale Integration at TI is Industry's Most Ambitious Effort," *Electronics*, May 26, 1977, pp 46-47
18. "Ultra High Speed Logic—More than Just a Promise in LSI," *Electronic Design*, Mar 29, 1977, pp 34-36

Responsible for definition and application support of bipolar arithmetic ICs at Monolithic Memories, Shlomo Waser's background includes digital computer design using microprogramming techniques and design of P/ROM programmers. He holds BSEE and MSEE degrees from San Jose State University, and is completing requirements for the Engineer degree at Stanford University.